

近似最近傍探索の多段階化による物体の高速認識

野口 和人[†] 黄瀬 浩一[†] 岩村 雅一[†]

[†] 大阪府立大学大学院工学研究科 〒599-8531 大阪府堺市中区学園町 1-1
E-mail: noguchi@m.cs.osakafu-u.ac.jp, {kise,masa}@cs.osakafu-u.ac.jp

あらまし SIFT などの局所記述子の最近傍探索によって物体認識を行う場合には、一画像あたりの特徴ベクトルの数が膨大になるため、最近傍探索の効率が重要となる。本稿では、「認識に必要な最近傍探索の精度は画像によって異なる」という観点から処理を削減した効率的認識法を提案する。具体的には、近似最近傍探索に基づく識別器を多段階に縦列接続することにより、認識に用いる近似の程度を画像に応じて変更し、大幅な効率化を実現する。一万画像のデータベースを用いた実験の結果、処理時間を、多段階化を行わなかった場合の約 1/10、ANN や LSH を近似最近傍探索の手法として用いた場合の約 1/40 に削減でき、例えば、認識率 98%、処理時間 1ms/query を達成できることがわかった。また、8.6%のリジェクトを行うことによって、誤認識率を 0%に抑えることもできた。さらに、十万画像を用いた実験から、高いスケーラビリティがあることもわかった。

キーワード 物体認識, SIFT, PCA-SIFT, 近似最近傍探索, 多段階化, リジェクト

Efficient Recognition of Objects by Cascading Approximate Nearest Neighbor Searchers

Kazuto NOGUCHI[†], Koichi KISE[†], and Masakazu IWAMURA[†]

[†] Graduate School of Engineering, Osaka Prefecture University
1-1 Gakuencho, Naka, Sakai, Osaka, 599-8531 Japan

E-mail: noguchi@m.cs.osakafu-u.ac.jp, {kise,masa}@cs.osakafu-u.ac.jp

Abstract For object recognition based on nearest neighbor search of local descriptors such as SIFT, it is important to keep the nearest neighbor search efficient to deal with a huge number of descriptors. In this report we propose a new method of efficient recognition based on the observation that the level of accuracy of nearest neighbor search for correct recognition depends on images to be recognized. The proposed method is characterized by the mechanism that multiple recognizers with approximate nearest neighbor search are cascaded in the order of the level of approximation so as to improve the efficiency by adaptively controlling the level to be applied depending on images. From experimental results with 10,000 images, we have confirmed that the proposed method is capable of achieving a recognition rate of 98% in 1 ms / query, which is 1/10 of the recognition time without the cascade, and 1/40 of the recognition time with conventional approximate nearest neighbor search such as ANN and LSH. In addition, a recognition error rate of the proposed method has been suppressed to 0% by allowing a rejection rate of 8.6%. Experimental results with 100,000 images show high scalability of the proposed method.

Key words Object recognition, SIFT, PCA-SIFT, Approximate nearest neighbor search, Cascade, Reject

1. はじめに

デジタルカメラやカメラ付き携帯電話の普及に伴って、カメラを情報入力機器としても利用したいという要望が高まっている。典型的な利用法としては、ポスターなどの物体を撮影して関連情報にアクセスすることが考えられる。このためには、画像中の物体を認識する必要がある。

物体認識の一手法として、SIFT(Scale-Invariant Feature Transform) [1] などによって得られる局所記述子の最近傍探索によるものがある。この手法には、特徴ベクトルが局所的に得られるために、隠れや変動に対して比較的口バストであるという利点がある。一方、得られる特徴ベクトルの次元が数十から数百、その数が画像あたり数百から数千と膨大なため、最近傍探索に必要な時間が膨大になるという問題点もある。

この問題は、最近傍探索の効率を向上させることによって解決可能である。一つのアプローチとして、最近傍探索に近似を導入し、効率化を図るものがある [2], [3]。これを物体認識に導入すると、効率を大幅に向上させることができる [4]。

近似最近傍探索を用いた物体認識では、近似の程度が認識率と効率をバランスするための重要なパラメータとなる。大幅な近似をすればするほど処理時間を削減できるが、近似をし過ぎると多くの特徴ベクトルに対して最近傍が求まらなくなり、結果として誤認識を引き起こしてしまう。ここでの問題は、誤認識を引き起こす近似の程度が画像によって異なる点である。大幅な近似を行っても認識できる「簡単な」画像がある反面、それでは誤認識となる「難しい」画像もある。固定的な近似によって一定の認識率を確保するには、近似の程度を認識の難しい画像に合わせる必要があり、効率向上の妨げとなっている。

そこで本稿では、近似の程度を画像に対して適応的に調節する効率的物体認識法を提案する。本手法では、近似の程度が異なる識別器を複数用意し、それらを大幅な近似のものからあまり近似のないものへと多段階に縦列接続する。これによって、簡単に認識できる画像は、前段の部分で大幅な近似の識別器によって高速に認識し、それでは認識できない画像に対してのみ、後段の部分で近似の少ない識別器によって時間をかけて精密に認識する。本手法の特徴は、多段階化する識別器の構成方法にある。後段の識別器では、近似の違いによる差分のみ、すなわち、それより前段の識別器で対象とならなかった特徴ベクトルのみを距離計算の対象とすることによって、最後段まで処理が進んでも、最後段の識別器を単独で用いる場合とほぼ同等の計算量しかかからないという利点を得ることができる。識別器として我々が先に提案した手法 (以降、野口らの手法とする) [4] を用いて 1 万画像のデータベースに対して実験を行った結果、認識率 98% を得るためにかかる処理時間を、多段階化を行わなかった場合の約 1/10, ANN, LSH といった従来の近似最近傍探索手法を用いた場合と比べて約 1/40 に削減できることがわかった。さらに、十万画像を用いた実験から、高いスケーラビリティがあることもわかった。

2. 最近傍探索に基づく物体認識

画像データベースに多数の画像が収められており、各々の画像は一つの物体を表すものとする。認識対象の画像 (以下、検索質問と呼ぶ) が与えられたとき、物体認識のタスクを、検索質問と同じ物体が写った画像をデータベースから検索することと定義する。本稿では、このようなタスクを最近傍探索によって達成する方法について考える。

最近傍探索を用いて物体認識を行う手法は、一つの画像を一つの特徴ベクトルで表現する方法 (ここでは、一画像一ベクトルと呼ぶ) と、一つの画像を複数の特徴ベクトルで表現する方法 (同様に、一画像多ベクトルと呼ぶ) に分類できる。以下では、検索質問の画像を Q 、データベース中の l 番目の画像を I_l と表す。また、 Q から得た特徴ベクトルを q_1, \dots, q_m 、 I_l から得た特徴ベクトルを p_{l1}, \dots, p_{ln_l} と表す。

まず、一画像一ベクトル ($m = 1, \forall l, n_l = 1$) で認識する方法

について述べる。この方法では、 q_1 に対して、データベースから最近傍を求め、 p_{l1} が得られたとすると、画像 I_l を回答とする。このような場合の最近傍探索法として、荒井らはデータの分布の偏りを利用して空間を分割する主成分木を提案している [5]。主成分木では、高次元のベクトルに対して、距離計算対象の絞込みと距離計算の打ち切りを効率よく行えるように空間を分割することによって処理を削減している。一画像一ベクトルの場合は、最近傍探索の精度が物体の認識率に直結するため、荒井らは近似最近傍探索ではなく、最近傍探索を行っている。

一方、一画像多ベクトル ($1 < i \leq m, 1 < j \leq n_i$) で認識する方法では、検索質問の特徴ベクトルが複数あるため、各々の最近傍探索の結果を統合しなければならない。最も単純な手法として、次の投票方式がある。 q_i に対してデータベース中から最近傍を求めて p_{lj} が得られたとすると、画像 I_l に 1 票を投じる。このような投票を Q から得られた m 個の特徴ベクトルに対して実行し、最終的に得票数が最大となった画像を認識結果として回答する。本稿では、 m, n_i は数百から数千の場合を考えるため、処理が膨大なものになる。

投票方式では、正解の物体と他の物体の得票数が逆転しなければ問題は生じない。そのため、一画像一ベクトルの場合と比べ、正確に最近傍を求める必要はない。このような観点から距離計算の対象の数を削減する方法として、bag of features という手法が提案されている [6]。この手法では、特徴ベクトルをそのまま扱うのではなく、ベクトル量子化して扱う。これにより、物体認識では数千から数万の量子化したベクトルに対して最近傍探索を行えばよく、処理が大幅に軽減される。しかし、量子化の影響により、認識率は最近傍探索を行う方法に比べ劣ることが多い。そのため本研究では、近似最近傍探索を用いて、認識率をあまり落とすことなく高速に認識することを考える。

3. 近似最近傍探索の従来法

提案手法の詳細について述べる前に、近似最近傍探索の従来法についてまとめておく。ここで取り上げる従来法は、比較実験に用いるものである。

最近傍探索で最も時間がかかるのは距離計算である。最近傍探索の高速化には、個々の距離計算自体を高速に行う方法と、距離計算の対象を効率よく絞り込む方法の 2 通りがある。近似最近傍探索では、主に後者の絞り込みを大胆に行うことで、処理時間を削減する。ところがその代償として、最近傍が距離計算の対象から外れてしまい、求まらない可能性がある。どの程度の近似が適切であるのかは、対象とするタスクに依存するため、近似最近傍探索の手法では、近似の程度を調節するためのパラメータを設けている。以下では、近似最近傍探索の代表的な手法として ANN と LSH について述べる。

ANN (Approximate Nearest Neighbor) [2] は、2 分木を用いて近似最近傍探索を高速に行う手法である。木のノードは、特徴空間を分割した hyperrectangle に対応しており、葉ノードには単一の特徴ベクトルが対応付けられている。ANN では、木構造の探索によって距離計算の対象となる特徴ベクトルを収集し、その中で距離が最短のものを近似最近傍探索の結果として

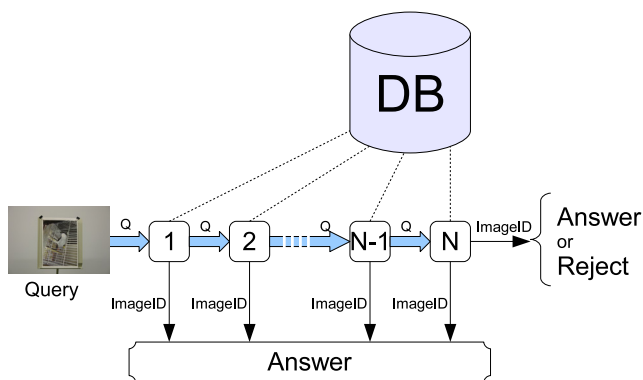


図 1 多段階化による認識

出力する．ANN には近似の程度を表すパラメータとして許容誤差 ϵ がある． ϵ が大きければ，より大幅な近似を行って対象となる特徴ベクトルを絞り込むため，処理時間が短縮できる．

LSH(Locality Sensitive Hashing) は，ハッシュ表を用いた近似最近傍探索の手法である [3]．LSH では，一つの特徴ベクトルを L 通りの k 次元ベクトルに変換し，各々に対応する L 個のハッシュ表に登録する．検索時には，検索質問の特徴ベクトル q を L 通りの k 次元ベクトルに変換し，すべてのハッシュ表を検索する．そして得られた特徴ベクトルの集合 P の中から q との距離が最小のものを結果とする．LSH では， k が一定ならば， L が小さいほど処理時間を短縮できるものの，真の最近傍も得られにくくなる．

4. 多段階化

4.1 構成と要件

一画像多ベクトルで画像を表現し，特徴ベクトルの近似最近傍探索と投票によって認識する場合，性能の限界は近似を行わない場合の認識率である．近似を行えばそれだけ高速化が実現できるが，一般に認識率は低下する．先に述べたように，このような近似の程度と認識率の関係は，認識対象の画像によって異なるため，認識率を保ちつつ処理時間を短縮するためには，近似の程度を適応的に調整する必要がある．

問題は，認識に必要な近似の程度を，認識の前に推定することが容易ではない点である．この問題に対処する一手法は，近似の程度が異なる多数の識別器を用意して，それらの識別器の出力を見ながら，適切なものを選択することであろう．

処理効率を保ちつつ複数の識別器を利用する方法としては，図 1 に示す多段階化が考えられる．ここで，1 から N の数字が付けられた矩形は識別器を表し，数字が若いほど大幅な近似を行うものとする．検索質問の画像から得た特徴ベクトルの集合は，まず 1 段目の識別器によって認識される．この段階で十分な証拠が得られれば，認識処理を打ち切って結果を回答する．一方，証拠が不十分ならば，特徴ベクトルの集合を，次段の，より小幅な近似の識別器にかけて再度認識する．処理を繰り返して最後の N 段まで到達しても十分な証拠が得られない場合には，最大得票数のものを回答するか，その画像についてはリジェクトするかのいずれかをとる．以上の処理によって，早い

段階で処理が打ち切られる画像については大幅な効率化が期待できると共に，必要に応じて時間をかけた認識が可能となる．

このような構成を採用する場合，要件となる事項は，

(1) 認識処理打ち切りの判定方法

(2) 「難しい」画像に対しても処理効率を保つ方法

の 2 点である．(1) については，少ない計算量で，なるべく正確に判定することが望まれる．(2) は，後段まで認識処理を繰り返す画像についても，処理効率を低下させないための方策である．理想的には，多段階化した識別器で s 段まで処理が進んだ場合の計算量が， s 段目と同じ近似の識別器を単独で使った場合と同等であればよい．以下，各々について述べる．

4.2 認識処理打ち切りの判定方法

認識誤りを引き起こす画像には，そもそも得票数が少ない，得票数がある程度得られる場合でも第 2 位の候補と得票数に開きが殆どない，という性質がある．これらの点に着目すると，得票数を用いた次のような簡便な判定方法が考えられる．1 位の得票数を V_1 ，2 位の得票数を V_2 とすると， $V_1 > t, rV_1 > V_2$ を同時に満たすならば，処理を打ち切って 1 位得票の画像を回答とする．ここで， t は得票数の閾値， r は 1 位と 2 位の得票数の比の閾値である．

4.3 「難しい」画像に対しても処理効率を保つ方法

近似の程度が異なる N 個の近似最近傍探索器 $1, \dots, N$ (以後，単に探索器と呼ぶ) を考える．探索器 $(s-1)$ の方が探索器 s よりも大幅に近似を行うとする．探索器 s を用いて，特徴ベクトル q_i に対して近似最近傍探索を行った結果，距離計算の対象として得られる特徴ベクトルの集合を $P_i^{(s)}$ とする．近似最近傍探索では，通常，大幅な近似であるほど，距離計算の対象となる特徴ベクトル数が少ないという性質がある．すなわち，すべての i と s に対して $|P_i^{(s)}| \geq |P_i^{(s-1)}|$ が成り立つ．

いま，これらの探索器に対して，次の 2 つの性質を考える．

[定義 1] (単調性) すべての i と s について，

$$P_i^{(s)} \supseteq P_i^{(s-1)} \quad (1)$$

が成り立つとき，近似最近傍探索器には単調性があるという．

[定義 2] (差分検索性) 近似最近傍探索器が差集合

$$P_i^{(s)} - P_i^{(s-1)} \quad (2)$$

を効率的に求められるとき，差分検索性があるという．

単調性を持つ探索器を用いて，図 1 の多段階識別器を構成する場合， s 段目では， $P_i^{(s)}$ ではなく，前段との差分 $P_i^{(s)} - P_i^{(s-1)}$ を距離計算の対象とすることが考えられる．このように構成すると，1 段目から s 段目までで距離計算の対象となった特徴ベクトルの和集合は，探索器 s を単独で用いた際の集合 $P_i^{(s)}$ と等しくなるため，距離計算の回数は同一となる．さらに，探索器が差分検索性を持つ場合には，多段階化を行っても計算量の増加を低く抑えることができる．

認識のプロセスを図 1 に沿って具体的に述べる．処理が $(s-1)$ 段目まで進んでいるときには，各特徴ベクトル q_i に

に対する暫定最近傍 \hat{p}_* が見つかった。従って、 s 段目では、 $p \in (P_s - P_{s-1})$ に対してのみ距離計算を行い、 \hat{p}_* よりも距離の近いベクトルが見つければ、それを新たに暫定最近傍 \hat{p}_* として、投票をやり直せばよい。

5. 提案手法

以上の概念に基づく手法を提案する。提案手法では、局所記述子として PCA-SIFT [7] を用いる。PCA-SIFT を用いる場合、ANN や LSH を用いた認識よりも高速な手法として、野口らの手法 [4] がある。この手法は、後述するように単調性ならびに差分検索性を持つため、多段階化にも好都合である^(注1)。そこで提案手法では、この手法を多段階化の識別器として用いて、物体認識法を構成する。

野口らの手法には、近似最近傍探索の最終段階で距離計算を行う手法（距離計算ありの手法）に加えて、距離計算を全く行わずに済ませる手法（距離計算なしの手法）がある。距離計算なしの手法は、距離計算ありの手法と比較し認識率は劣ものの、特徴ベクトルをメモリ上に保持しておく必要がないため、省メモリであるという利点がある。以下では、まず、距離計算ありの手法、距離計算なしの手法に共通のデータ登録について述べたあと、各々の手法、多段階化の方法について述べる。

5.1 データの登録

LSH では、用いるハッシュ表の数が多くなると大量のメモリを消費する。そこで野口らの手法では、メモリ量を削減するため、ハッシュ表を 1 つだけ使う。ハッシュ表に特徴ベクトルを登録する方法は次のとおりである。PCA-SIFT によって得られた d' 次元のベクトル $p = (p_1, \dots, p_{d'})$ の第 1 次元から第 d 次元 ($d \leq d'$) までを用い、

$$u_j = \begin{cases} 1 & \text{if } p_j \geq 0, \\ 0 & \text{otherwise,} \end{cases}$$

によって各次元を 2 値化し、ビットベクトル $u = (u_1, \dots, u_d)$ を作成する。そして、

$$H_{\text{index}} = \left(\sum_{i=0}^d u_i 2^i \right) \bmod H_{\text{size}} \quad (3)$$

によってハッシュのインデックスを求め、画像 ID をハッシュ表に登録する。ここで H_{size} は、ハッシュ表のサイズである。距離を用いる場合には、画像 ID のほか、 p そのものを登録し、検索時の距離計算に用いる。

登録時に衝突が生じた場合は、チェーン法により複数の画像 ID をリストとして登録する。このとき、リストが長くなりすぎると、距離計算のコストがかかりすぎるといった問題が生じる。野口らの手法では、リスト長 n に対する閾値 c を設け、 $n > c$ を満たすとリスト全体をハッシュ表から削除する。同じハッシュ値を持つ特徴ベクトルが多いということは、その特徴ベクトルが画像の識別にあまり寄与しないことを意味する。従って、削

除をしても影響は比較的少ないと考えられる。これは、bag of features での stop word と同じ考え方である。

以上の処理を、データベースに登録するすべての特徴ベクトルに対して施すことにより、データの登録は完了する。

5.2 検索

5.2.1 距離計算ありの手法

次に距離計算ありの手法の検索について述べる。この手法では、検索質問 Q から得た各特徴ベクトル q_i に対して、登録時と同様にインデックスを計算することで、ハッシュ表から特徴ベクトルを検索する。得られた特徴ベクトルの集合を P_i とすると、次に P_i の中から q_i の最近傍となる特徴ベクトル \hat{p}_* を求める。そして、 \hat{p}_* に対応する画像 ID に投票する。

検索質問画像とデータベースの画像が同じではないため、特徴ベクトルの各次元の値は変動したものになる。このとき、異なったビットベクトルに変換されると、検索が不可能になるという問題が生じる。野口らの手法では、値の許容変動幅 e をパラメータとして、変動への対処を施す。具体的には、 $q = (q_1, \dots, q_d)$ とするとき、 $|q_j| \leq e$ を満たす次元 j に対しては、 u_j だけではなく $u'_j = (u_j + 1) \bmod 2$ (0 ならば 1 , 1 ならば 0) も用いて、特徴ベクトルを検索する。例えば、ビットベクトル $u = (0, 0, 0, 1)$ の第 2 次元がこの処理の対象となったとすると、 $u' = (0, 1, 0, 1)$ も用いて検索する。ただし、このような「両方試す」という処理を制限なく導入すると、膨大な計算時間が必要となってしまうため、 u'_j を適用する次元数を一定値 b 以下に制限する。これによって、最も多い場合でも、用いるビットベクトルの数を 2^b に抑えている。 $|q_j| \leq e$ を満たす次元の数が b を上回る際には、次元のインデックスが小さいものから b 個を採用する。

5.2.2 距離計算なしの手法

距離計算なしの手法では、検索質問の特徴ベクトル q に対して上記のような距離計算を施して近似最近傍を求めるのではなく、ハッシュ表から得た特徴ベクトルの集合 P に属するすべての特徴ベクトルに対して投票処理を施す。処理のパラメータは、距離を用いない方法と同様、特徴量の許容変動幅 e 、変動に対処する次元の数 b の 2 つである。

5.2.3 b による多段階化

野口らの手法のパラメータは b, c, d, e の 4 つである。提案手法では、このうち b を変更することで近似の程度を調整する。具体的には、第 s 段では $b = s - 1$ とした識別器を用いる。野口らの手法は、 b の増加に伴ってハッシュ表のアクセスに用いるインデックスが増加するだけである。そのため、単調性だけではなく差分探索性も満たす。

なお、距離計算なしの手法では、各段の処理で、暫定最近傍 \hat{p}_* を更新しつつ投票するのではなく、得られた差集合 $P_i^{(s)} - P_i^{(s-1)}$ に属する特徴ベクトルすべてに対して投票する。

6. 実験

6.1 実験条件

実験では、近似最近傍探索の従来法である ANN, LSH^(注2)に

(注1): 単調性は ANN や LSH でも満たされる。ANN では ϵ の値を段階的に変更し、LSH では探索するハッシュ表の数 L を段階的に増やせばよい。

(注2): ANN としては <http://www.cs.umd.edu/~mount/ANN/>, LSH として



(a) A の例 (b) B の例 (c) C の例

図 2 登録画像の例

表 1 データベースに含まれる画像数

データベース	DB1	DB2	DB3	DB4
登録画像数	5,000	10,000	50,000	100,000
内訳	A	1,667	3,100	3,100
	B	1,667	3,450	18,500
	C	1,666	3,450	28,400

加え、野口らの手法を用いて識別器を構成し、提案手法（野口らの手法の多段階化版）と比較した。

実験には以下に述べる画像データベース、検索質問画像を用いた。局所記述子としては、PCA-SIFT のサイト^(注3)で提供されるものを用いた。ハッシュ表のサイズは $H_{size} = 2^d$ とした。以下に示す処理時間は、検索質問の画像 1 枚あたりの認識に要した時間を表す。ただし、特徴ベクトルの抽出に必要な時間は含めない。使用計算機は、CPU が AMD Opteron 2.8GHz、メモリ 32GB のものである。

6.1.1 画像データベース

実験に用いた画像について説明する。まず、収集方法の異なる A, B, C の 3 種類のデータセットを準備した。A は、Google のイメージ検索を用いて収集した 3,100 枚の画像である。検索キーワードとしては、“ポスター”、“雑誌”、“表紙”などを用いた。図 2(a) に例を示す。B は PCA-SIFT のサイトで公開されている画像であり、画像数は 18,500 枚である。このデータは主に自然写真や人物の写真などで構成されている。図 2(b) に例を示す。C は、写真共有サイトの flickr において “animal”、“birthday”、“food”、“japan” などのタグにより収集した 78,400 枚の画像からなる。主に図 2(c) に示すような物体や自然の写真、人物の写真などを含む。なお、収集の際には、600×600 pixel 以下のサイズの画像は除外し、画像の長辺が 640 pixel 以下になるように縮小した。また、特徴ベクトルが 100 個以下の画像も除外した。画像の一辺の長さの平均は A, B, C それぞれ 498, 612, 554 pixel であった。

次に、A, B, C の画像を用いて、表 1 に示した画像数からなる 4 種類のデータベース DB1~DB4 を作成し実験に用いた。ここで、大きいデータベースは、小さいデータベースをその一部として含む。なお、DB2(1 万枚)からは、一画像あたり平均 2,069 個の特徴ベクトルが抽出された。



(a) 撮影角度 90° (b) 撮影角度 75°



(c) 撮影角度 60° (d) 撮影範囲一部

図 3 検索質問の例

6.1.2 検索質問画像

検索質問としては、データベースに対応する画像のあるものとなないものの 2 種類を作成した。前者については、データベースに含まれる画像の中から、データセット A, B, C のそれぞれから 100, 200, 200 枚の合計 500 枚を無作為に選択した。後者については、画像データベースには含まれない画像を 199 枚用意した^(注4)。次に、これらを A4 の用紙に印刷し、カメラを用いて撮影した。得られた画像の例を図 3 に示す。図に示すとおり、紙面全体が写る配置で、紙面に対するカメラの光軸の角度 θ を 90°, 75°, 60° に変化させた。また、角度を 90° として紙面の一部を撮影した。その結果、1 枚の紙面に対して、合計 4 通りの画像を得た。さらに、撮影した画像を 512 × 341 pixel に縮小し、PCA-SIFT により特徴ベクトルを求めた。その結果、画像 1 枚あたり平均 612 個の特徴ベクトルが得られた。

6.2 リジェクトをしない場合

まず、対応する画像がデータベースにある検索質問のみを用いて、実験を行った。

6.2.1 近似最近傍探索の精度と画像の認識率の関係

予備実験として、DB2(1 万枚)を用いて、画像を認識するために必要な最近傍探索の精度を調べた。具体的には、多段階化なしの手法 (ANN, LSH, 距離計算ありの手法) について、パラメータをさまざまに変え、近似最近傍探索の精度と画像の認識率の関係を計測した。近似最近傍探索の精度とは、近似最近傍探索によって真の最近傍が求めた割合である。

結果を図 4 に示す。この結果から、近似最近傍探索の精度が 100% から 20% あたりまでは、近似最近傍探索の精度が減少しても、認識率はほとんど減少しないことがわかる。これは、間違っただけの画像に投票されても、正解の画像と他の画像の得票数が逆転するまでには至らないためと考えられる。また、近似最近傍探索の精度と認識率には、手法に依存しない関係があることも伺える。

は <http://www.mit.edu/~andoni/> で提供されているソースコードを用いた。

(注3): <http://www.cs.cmu.edu/~yke/pcasift/>

(注4): 当初は 200 枚用意したが、うち 1 枚が悪いもかけずデータベースに含まれていることが判明したため 199 枚となった。

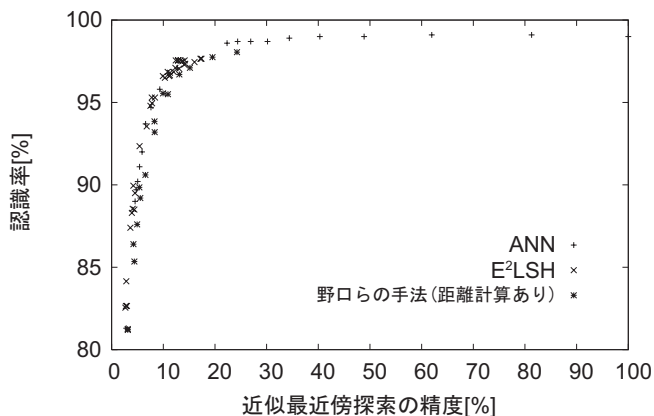


図 4 近似最近傍探索の精度と画像の認識率

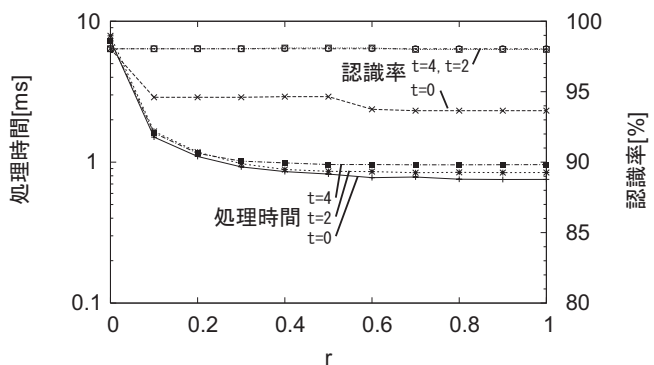


図 5 距離計算あり: r, t と認識率, 処理時間の関係

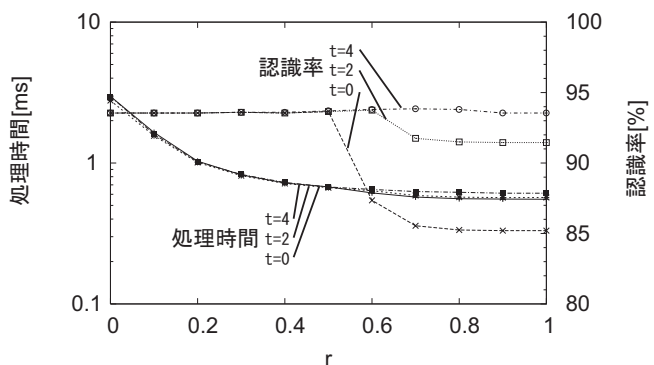


図 6 距離計算なし: r, t と認識率, 処理時間の関係

6.2.2 認識処理打ち切りのパラメータと認識率, 処理時間の関係

DB2(1万枚)を用いて, 認識処理打ち切りに関するパラメータについて調べた. まず, 距離計算ありの手法を用いて, r, t と認識率と処理時間の関係について調べた. 具体的には, $b = 10, c = 5, d = 28, e = 200$ とし, $t = 0, 2, 4$ について r を 0 から 1 まで 0.1 刻みに変化させ, 処理時間と認識率を調べた. 結果を図 5 に示す. t は最低得票数の閾値であり, $t = 0$ という特殊な場合を除くと, 認識率には大きな変化は見られず, 処理時間は r を大きくするにつれて減少していることがわかる.

同様に, 距離計算なしの手法を用いて, r, t と認識率と処理時間の関係について調べた. $b = 6$ とし, 他のパラメータについては, 距離計算ありの手法と同一とした. 結果を図 6 に示す.

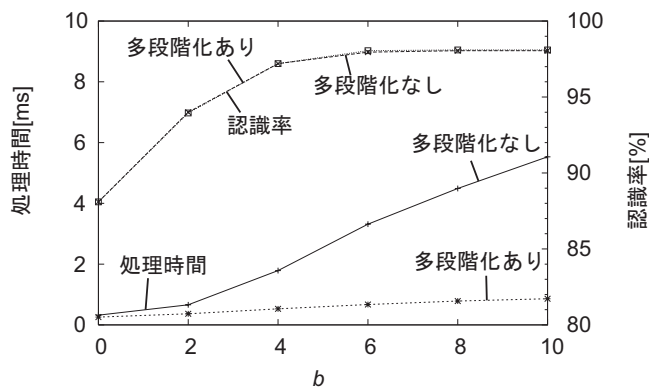


図 7 距離計算あり: b と認識率, 処理時間の関係

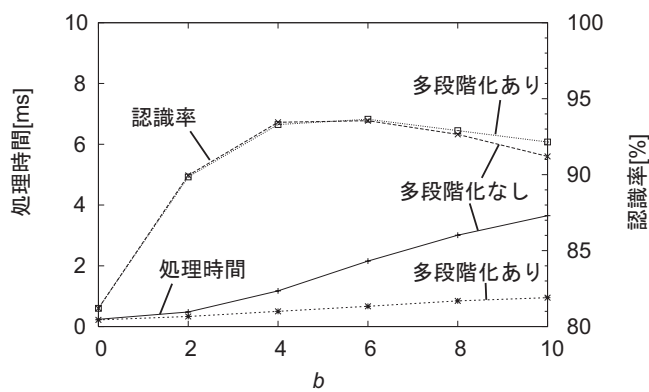


図 8 距離計算なし: b と認識率, 処理時間の関係

$r = 0.5$ 以下の場合には, 認識率が低下することなく, 処理時間を削減できていることがわかる.

6.2.3 多段階化の段数と認識率, 処理時間の関係

次に, DB2(1万枚)を用いて, 多段階化の段数 N を変更し, 多段階化の効果を検証した. まず, 距離計算ありの手法を用いて, $b (= N - 1)$ と認識率と処理時間の関係について調べた. $c = 5, d = 28, e = 200, r = 0.5, t = 2$ とし, b を変化させた結果を図 7 に示す. 多段階化を行う場合は, 行わない場合と比較し, ほとんど認識率が低下せずに, 処理時間を大幅に削減できることがわかった. また, 多段階化の段数が多くなるにつれて, 処理時間削減の効果が大きくなっていることがわかった. 同様に, 距離計算なしの手法を用いて, b と認識率と処理時間の関係について調べた. $c = 3, d = 28, e = 200, r = 0.5, t = 2$ とし, b を変化させた結果を図 8 に示す. 距離計算なしの手法でも, 処理時間を削減できることがわかった.

6.2.4 各段の認識器で出力される割合

DB2(1万枚)を用いて, 各検索質問について何段目で認識打ち切りが行われているかを調べた. パラメータは, 先ほどの実験の結果で多段階化による処理時間の削減効果が高かった, $b = 10, c = 5, d = 28, e = 200, r = 0.5, t = 2$ とした. 距離計算ありの手法による結果を図 9 に示す. 図上部の枠内に, 多段階化の構成を示す. 各段で認識打ち切りが行われ, 回答が出力された検索質問の枚数を集計した. 最終段 (11 段目) では, 認識処理打ち切りの判定に合格しなかった検索質問についても, 強制的に回答が出力される. そのため, 認識処理打ち切りの判

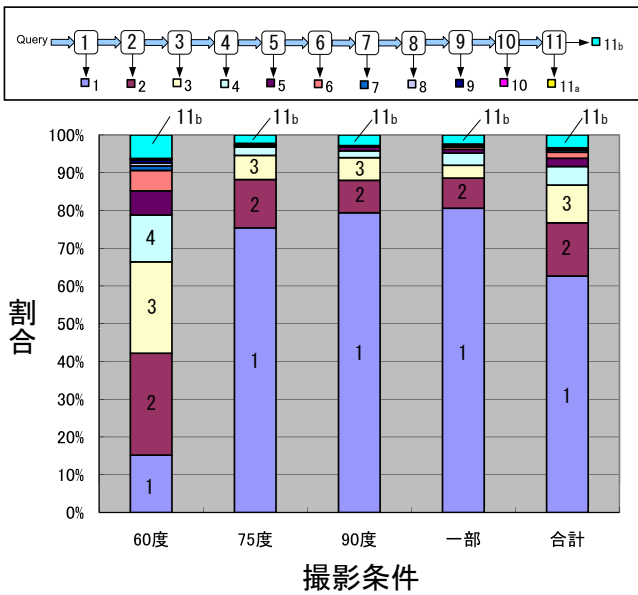


図 9 距離計算あり:各段での出力の割合

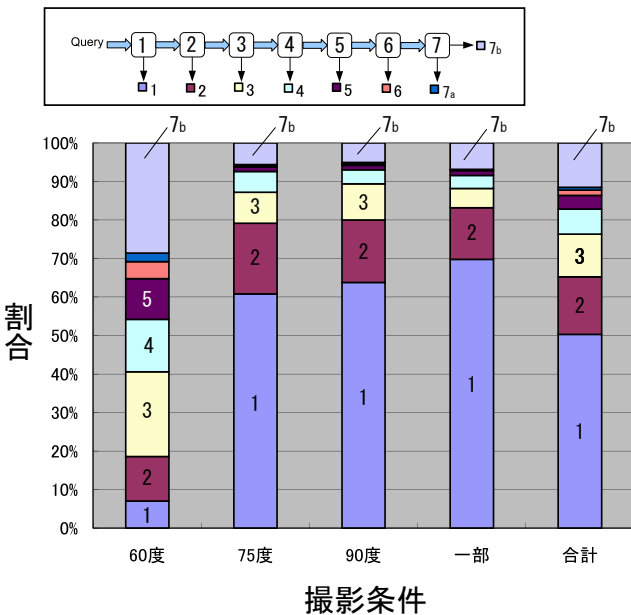


図 10 距離計算なし:各段での出力の割合

定に合格したものを 11_a , 不合格のものを 11_b として集計した .

結果から , 検索質問全体の 63% が一段目で回答されている . 撮影条件を比較すると , 90 度 , 75 度 , 60 度と撮影条件が厳しくなるほど , 一段目で回答した割合が減少している . このことから , 近似の程度を画像に応じて変更し , 認識の簡単な画像に対しては , 時間をかけずに認識していることがわかる .

同様に , 距離計算なしの手法を用いた結果を図 10 に示す . パラメータは , 先ほどの実験の結果より $b = 6, c = 5, d = 28, e = 200, r = 0.5, t = 2$ とした . 距離計算なしの手法は , 距離計算ありの手法と比べて認識率が劣るため , 一段目で認識できる割合は減少しているものの , 近似の程度を画像に応じて変更できていることがわかる .

6.2.5 各手法の比較

各手法の特徴を比較するため , パラメータをさまざまに変え ,

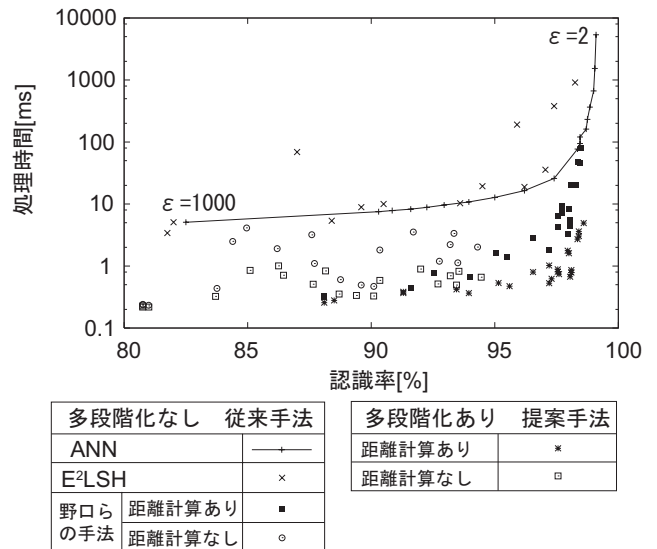


図 11 リジェクトをしない場合の各手法の比較

認識率と処理時間の関係を描いたグラフを図 11 に示す . ANN で許容誤差 ϵ を 2 から 1000 まで変化させたものを線で描き , 評価の基準とした . 右にプロットされているものほど認識率が高く , 下にプロットされているものほど処理時間が短い . そのため , 右下にプロットされているものほど優れているといえる . LSH は , ほぼ ANN の線を越えることはなかった . 野口らの手法では , 最大認識率は ANN に及ばないものの , 認識率が 98% 以下の場合には , 同じ認識率を , ANN に比べて 1/10 から 1/40 程度の処理時間で実現できている . 提案手法では , 多段階化を行うことで , 処理時間が野口らの手法の 1/5 程度にまで削減されている .

6.2.6 スケーラビリティ

次に , 登録画像数と認識率と処理時間の関係について調べた .

まず , 距離計算ありの手法を用いて , 登録画像数を 5000 枚 (DB1) から 10 万枚 (DB4) まで変化させ , 実験を行った .

パラメータは , DB4(10 万枚) を用いた場合に , 良い認識率が得られた $c = 10, d = 28, e = 400, r = 0.5, t = 2$ を用いた . 登録画像数を変化させた結果を図 12 に示す . 多段階化を行わない場合と比較すると , 登録画像数によらず , ほぼ同じ割合で処理時間が削減できていることがわかる .

同様に , 距離計算なしの手法を用いて , 登録画像数と認識率と処理時間の関係について調べた . パラメータは , 10 万枚の画像のデータベースを用いた場合に , 良い認識率が得られた $c = 10, d = 28, e = 400, r = 0.5, t = 2$ を用いた . 登録画像数を変化させた結果を図 13 に示す . 距離計算なしの手法でも , 登録画像数によらず , ほぼ同じ割合で処理時間が削減できている .

なお , DB4(10 万枚) を用いた場合のメモリ使用量は , 距離計算ありの手法で約 20GB , 距離計算なしの手法で約 6.4GB であった .

6.3 リジェクトをする場合

最後に , DB2(1 万枚) を用いたリジェクトをする場合の実験結果について述べる . 結果の評価尺度を次の様に定める . まず , 対応する画像のある検索質問に対しては , 認識率 C_1 , 誤認識率

表 2 リジェクトをする場合の結果

パラメータの決定基準 (学習サンプルに対して)		手法	対応する画像のある検索質問				対応する画像のない検索質問		
			C_1 [%]	E_1 [%]	R_1 [%]	処理時間 [ms]	E_2 [%]	R_2 [%]	処理時間 [ms]
基準 A	$E_1 = 0, E_2 = 0$	距離計算あり	91.4	0.0	8.6	2.6	0.0	100	17.2
	処理時間 20ms 以下, $\min R_1$	距離計算なし	69.4	0.05	30.55	1.9	0.38	99.62	5.1
基準 B	$\min(E_1 + E_2 + R_1)$	距離計算あり	95.75	0.0	4.25	2.6	1.25	98.75	18.7
	処理時間 20ms 以下	距離計算なし	86.30	0.05	13.65	1.0	2.26	97.74	6.0

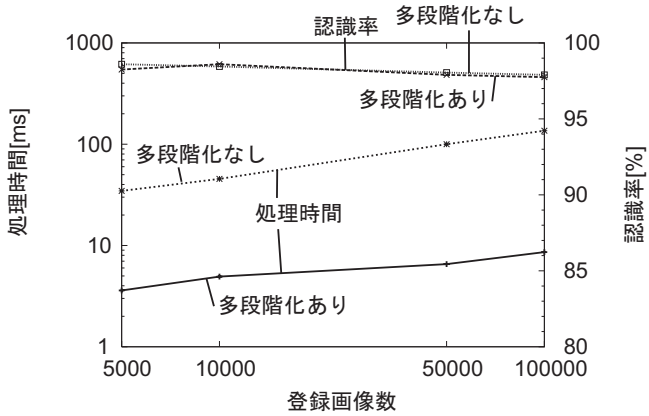


図 12 距離計算あり:登録画像数と認識率, 処理時間の関係

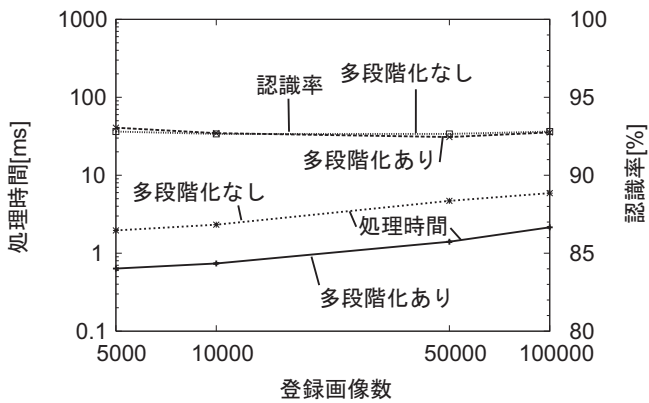


図 13 距離計算なし:登録画像数と認識率, 処理時間の関係

E_1 , リジェクト率 R_1 ($C_1 + E_1 + R_1 = 1$) とする. 対応する画像のない検索質問に対しては, 誤認識率 E_2 , リジェクト率 R_2 ($E_2 + R_2 = 1$) とする. 実験には, 10-fold cross validation を用いた. 学習サンプルに対して, 以下の二通りの基準を用いてパラメータを求め, テストサンプルに適用した.

基準 A $E_1 = 0, E_2 = 0$, 処理時間 20ms 以下という条件の下で R_1 が最小

基準 B 処理時間 20ms 以下という条件の下で $E_1 + E_2 + R_1$ が最小

パラメータについては, 距離計算ありの手法, 距離計算なしの手法の両方で, $b = 5, 10, 15, c = 2, 5, 10, 100, d = 20, 24, 28, e = 200, 400, 600, r = 0.2, 0.4, 0.6, t = 4, 8, 12$ のすべての組み合わせについて試した.

結果を表 2 に示す. 基準 A でパラメータを設定した場合, 距離計算ありの提案手法では, リジェクト率 R_1 が 8.6% の場合に誤認識率 E_1, E_2 を 0% とすることができた. 基準 B の場合には, 誤認識率 E_2 が 1.25% に増加したものの, 認識率 C_1 を増

加させることができた. 一方, 距離計算なしの提案手法では, 距離計算ありの手法に及ばない結果となった.

処理時間については, 対応する画像のない検索質問の方が, 3 から 7 倍程度長くなった. これは, 多段階化によって, 対応する画像のない検索質問のほとんどは, 最終段まで処理されてリジェクトされるのに対し, 対応する画像のある検索質問は, 最終段まで到達せずに回答が出力されているためである.

以上のように, 提案手法では 5~9% 程度のリジェクトによって, 処理時間を犠牲にせず, 誤認識率を極めて低く抑え得ることがわかった.

7. ま と め

本稿では, 近似最近傍探索による識別器を, 近似の程度を変化させて多段階に接続することにより, 高速に物体を認識する手法を提案した. 1 万画像を対象とした実験の結果, ANN や LSH といった従来の近似最近傍探索手法を用いる場合と比較して, 認識率 98% を 1/40 程度の処理時間で得ることができた. また, 1 割程度のリジェクトを行うことで, 誤認識を 0% に近づけることができた.

今後の課題には, 平面物体以外への適用などがある.

謝辞 本研究の一部は日本学術振興会科学研究費補助金 (基盤研究 (B)19300062) の補助による.

文 献

- [1] D. Lowe: "Distinctive image features from scale-invariant keypoints", International Journal of Computer Vision, **60**, 2, pp. 91-110 (2004).
- [2] S. Arya, D. Mount, R. Silverman and A. Y. Wu: "An optimal algorithm for approximate nearest neighbor searching", Journal of the ACM, **45**, 6, pp. 891-923 (1998).
- [3] M. Datar, N. Immorlica, P. Indyk and V. Mirrokni: "Locality-sensitive hashing scheme based on p-stable distributions", Proc. of the 20th Annual Symposium on Computational Geometry, pp. 253-262 (2004).
- [4] 野口, 中居, 黄瀬, 岩村: "特徴ベクトルの近傍探索と物体認識の効率に関する実験的検討", 電子情報通信学会技術研究報告, PRMU2006-68, pp. 57-64 (2006).
- [5] 荒井, 武本, 加藤, 和田: "階層的固有空間による高次元最近傍探索の高速化", 画像の認識・理解シンポジウム (MIRU2006), pp. 291-297 (2006).
- [6] J. Sivic and A. Zisserman: "Video Google: a text retrieval approach to object matching in videos", Proc. of ICCV2003, pp. 1470-1477 (2003).
- [7] Y. Ke and R. Sukthankar: "PCA-SIFT: A more distinctive representation for local image descriptors", Proc. of CVPR2004, Vol. 2, pp. 506-513 (2004).